This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problems Mailbox.

THIS PAGE BLANK (USPTO)

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 508 633 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: **92302428.5**

(51) Int. Cl.[5]: **G06F 9/44**, G06F 3/033

(22) Date of filing: **20.03.92**

(30) Priority: **12.04.91 US 685236**

(43) Date of publication of application:
14.10.92 Bulletin 92/42

(84) Designated Contracting States:
DE FR GB

(71) Applicant: **Hewlett-Packard Company**
3000 Hanover Street
Palo Alto, California 94304(US)

(72) Inventor: **Hunt, William L.**
8307 Bruns Drive
Fort Collins, Colorado 80525(US)

(74) Representative: **Colgan, Stephen James et al**
CARPMAELS & RANSFORD 43 Bloomsbury Square
London WC1A 2RA(GB)

(54) Dynamic user panels.

(57) Disclosed is a system that allows a programmer or user to dynamically create a user panel (706, 708), including selecting the icons (808), for an iconic network program, or part of an iconic network program, within an iconic programming environment. The programmer or user may select a show-on-execute option (714) for the panel which causes the panel to be displayed when any one of the icons within the panel is processed by the iconic programming system (1006, 1016). The panel may also be secured by removing all panel options that would allow any change to the panel or the iconic network program (712).

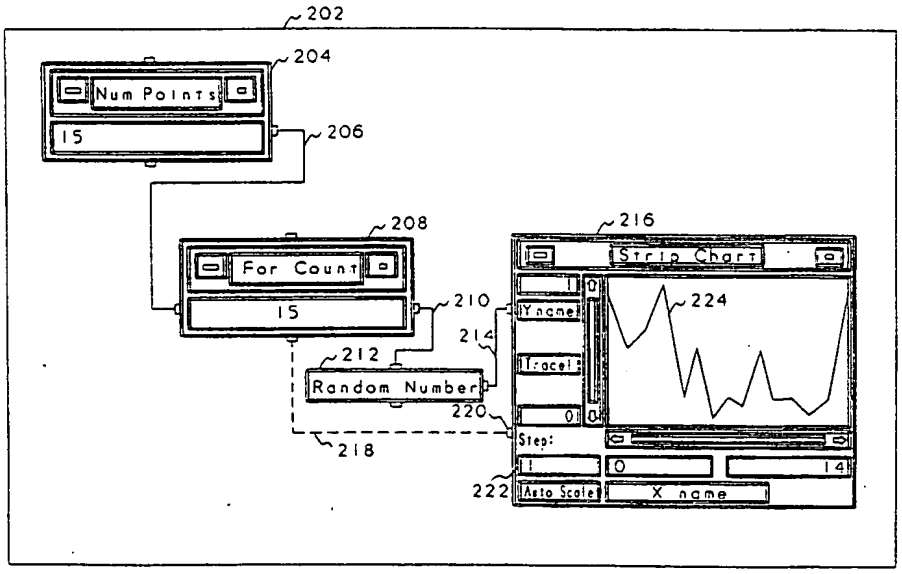

FIG. 2

EP 0 508 633 A2

BNSDOCID: <EP 0508633A2>

FIELD OF THE INVENTION

This invention relates to computer systems and more particularly to Iconic Programming Systems. Even more particularly, the invention relates to dynamically creating user panels in such a system.

BACKGROUND OF THE INVENTION

An iconic programming system is a "programming-less" environment where programming is done by connecting graphical images of tools (icons), together with connecting lines, to form a directed graph and create an iconic network which represents a software program. The iconic programming system may be used in research and development test environments, where several different electronic instruments are connected to test a system or device. Programming such a system requires instructions to cause the various instruments to perform desired functions in order to operate as a system.

When an iconic programming system is used, each instrument will be represented by a graphical icon, also called a graphical object, and the connections between the instruments are represented by lines between these graphical images. In addition to graphical icons representing instruments in such a system, graphical icons are provided for programming functions, for example looping, IF-THEN statements, etc. By combining instrument and programming icons, a user can create an iconic network involving the programmed operation of several instruments.

An iconic network program can be large and/or complicated. Often the user of the program does not need to know the details of all icons within the program, but does need to see certain icons such as data input icons and data display icons. Also, some icons need only display when certain events occur. This provides a way of displaying information only when the information is needed.

Prior art iconic programming environments, such as the environment disclosed in U.S. Patents 4,901,221 issued Feb. 13, 1990 to Kodosky, et al. and 4,914,568 issued April 3, 1990 to Kodosky et al. do not allow a user to select elements for a user panel and includes all Input/Output elements on the panel. Also, the panels in this system do not allow non-input/output elements on a panel.

There is a need in the art then for a system that will provide a way for a programmer in an iconic system to define a user panel. There is further need for such a system to allow a panel to be defined to only be visible when an icon is processed. A still further need is to allow a user to select the elements that will be included in the panel. The present invention meets these needs.

Various features and components of an iconic network system are disclosed in U.S. patent applications:

(A) Application serial No. 07/483,478 filed 02/22/90 for PROGRAMMING ESCAPE FROM AN ICONIC SYSTEM of Bailey, Beethe, Wolber, and Williams;

(B) Application serial number 07/537,550 filed Jun. 13, 1990 for PROCESSING METHOD FOR AN ICONIC PROGRAMMING SYSTEM of Beethe;

(C) Application serial No. 07/661,936 filed Feb. 28, 1991 for LINE PROBE IN AN ICONIC PROGRAMMING SYSTEM of Wolber;

(D) Application serial No. 07/668,286 filed Mar. 12, 1991 for INPUT AND OUTPUT CONSTRAINTS ON DEVICES IN AN ICONIC PROGRAMMING SYSTEM of Wolber and Beethe; and

(E) Application serial No. 07/668,285 filed Mar. 12, 1991 for A SYSTEM FOR ENTERING AND MODIFYING VARIABLE NAMES FOR TERMINALS OF ICONS IN AN ICONIC PROGRAMMING SYSTEM of Wolber and Beethe.

each of which is hereby specifically incorporated by reference for all that is disclosed therein.

SUMMARY OF THE INVENTION

It is an aspect of the present invention to provide a system that allows an iconic programmer within an iconic programming system to dynamically create a user panel for an iconic network.

It is another aspect of the invention to provide such a system that will optionally display the user panel when an icon within the panel is processed.

A further aspect of the invention is to allow the panel to be secured to prevent further change to the panel or its network program.

The above and other aspects of the invention are accomplished in a system that allows a programmer or user of the system to dynamically create a user panel for an iconic network program. The user selects the icons from the network that are to be included on the panel and then selects an add to panel function to place the icons on a panel. The system displays the panel after adjusting the icon locations so that they fit within the panel. The programmer or user is then allowed to further move the icons to a desired location.

The programmer or user may select a show-on-execute option for any panels for user objects (subprograms). This will cause these panels to display automatically if one of the icons on the panel is processed by the iconic programming system. The programmer or user may also secure the panel by removing all panel options that would allow any change to the panel or its iconic network

BNSDOCID: <EP 0508633A2>

program.

BRIEF DESCRIPTION OF THE DRAWINGS

The above and other aspects, features, and advantages of the invention will be better understood by reading the following more particular description of the invention, presented in conjunction with the following drawings, wherein:

Fig. 1 shows a block diagram of a computer system incorporating the present invention;

Fig. 2 shows an example iconic network;

Fig. 3 shows a user panel created from the iconic network of Fig. 2;

Fig. 4 shows a more complicated example iconic network;

Fig. 5 shows a user panel created from the iconic network of Fig. 4;

Fig. 6 shows an example of a user panel containing a show-on-execute user panel created from the network of Fig. 4;

Fig. 7 shows a flowchart of the top level process used by the iconic programming system to interact with the user;

Fig. 8 shows a flowchart of the create user panel process;

Fig. 9 shows a flowchart of the secure user panel process; and

Fig. 10 shows a flowchart of the execution process and illustrates the show-on-execute process.

DESCRIPTION OF THE PREFERRED EMBODIMENT

The following description is of the best presently contemplated mode of carrying out the present invention. This description is not to be taken in a limiting sense but is made merely for the purpose of describing the general principles of the invention. The scope of the invention should be determined by referencing the appended claims.

Fig. 1 shows a block diagram of a computer system incorporating the present invention. Referring now to Fig. 1, a computer system 100 contains a processing element 102 which connects to the other components of the system through a system bus 104. A keyboard 106 allows a user to input textual data to the system, and a mouse 110 allows a user to input graphical data to the system. A graphics display 108 allows the system to output text and graphical information to the user. A disk 112 is used by the system to store the software of the iconic programming system environment, as well as the user-defined iconic network. An instrument bus 116, typically an IEEE 488 bus, is used to allow the iconic programming system to communicate with test instruments. Memory 114 is used to contain the software and user data of the iconic programming environment. A multi-tasking operating system 120 can have a plurality of tasks, also called processes, here illustrated by task 122 and task 124. Task 122 is shown containing the iconic programming process of the present invention.

Fig. 2 shows an example iconic network. Referring now to Fig. 2, a display screen 202 contains a numeric input icon 204 which is used to define a number of points, in this example 15, which will be displayed on a strip chart icon 216. The number of points is sent over a connecting line 206 to a count icon 208. The count icon 208 will count from zero to the number of points received on the connecting line 206. Each time the count icon 208 increments the count value by one, a signal will be sent over connecting line 210 to a random number icon 212. When the random number icon 212 receives the signal on connecting line 210, the random number icon 212 outputs a random number over a connecting line 214 which is input as the Y value to the strip chart 216.

After all 15 random numbers have been output by the random number icon 212, a sequence signal is sent from the count icon 208 over dashed connecting line 218 to autoscale input 220 of the strip chart 216. This signal will cause the strip chart to rescale and redisplay a waveform 224.

The iconic programmer who created the system illustrated in Fig. 2 needs to see the complete display of Fig. 2. However, to use the iconic network, a user of the iconic network of Fig. 2 would not need to see all the details that are shown in Fig. 2. A user would only need to see the number input icon 204 and the strip chart icon 216. The details involved in the count icon 208, the random number icon 212, and the connecting lines would not be necessary just to use the iconic network. In fact, showing these extra details can be quite confusing on a large iconic network. Therefore, the system of the present invention allows an iconic programmer to select certain icons to be displayed on a user panel, thus hiding unnecessary details for a user of the system.

Fig. 3 shows a user panel created from the iconic network of Fig. 2. As described above, a user of the iconic network of Fig. 2 would only need to see the number input icon 204 and the strip chart 216. Therefore, the user panel of Fig. 3 only shows these icons, and hides the other details which are unnecessary to a user of the network. Referring now to Fig. 3, a screen 302 shows the user panel created from the network of Fig. 2. The numeric input icon 204 is shown to the side the strip chart 216. Because of the additional space provided by eliminating unnecessary icons on the display, the strip chart 216 can be expanded in size to be more visible to the user of the system.

BNSDOCID: <EP 0508633A2>

Fig. 4 shows a more complicated example of an iconic network and will be used to illustrate user panels that contain elements that display only when the underlying icon is processed. Referring now to Fig. 4, a display screen 402 contains a numeric input icon 404, which has been used to input the value 15. The numeric icon 404 sends the value to a count icon 406 which will count from zero to the value input from the numeric input icon 404. Each time the count icon 406 increments its counter, it sends a signal to a random number icon 408 which outputs a random number to the Y input 409 of a strip chart 410. After 15 random numbers have been sent from the random number icon 408 to the Y input 409, the count icon 406 outputs a sequence signal over a connecting line 418 which is sent to an autoscale input 411 of the strip chart 410. These icons and connections will cause the strip chart 410 to display a random waveform 420 in the same manner as Fig. 2.

In the iconic network of Fig. 4, however, additional icons are used to accumulate the information in a different manner. The random number output by the random number icon 408 is sent to the input of a sliding collector 414 in parallel with being sent to the Y input 409 of the strip chart 410. The sliding collector 414 will collect each of the random numbers and store them in an array. This array will be sent to an actual values icon 416, which has been created as a user object. The sequence output of the count icon 406, which is sent to the step input 411 of the strip chart 410, is also sent in parallel to a show values icon 412. This icon will be used to trigger a user panel which will display the actual values sent to the strip chart 410. A sequence output of the show values icon 412 is sent over a connecting line 424 to a sequence input of the actual values icon 416.

Fig. 5 shows a user panel created from the iconic network of Fig. 4. Referring now to Fig. 5, a display screen 502 contains the numeric input icon 404, the strip icon 410, and the show values icon 412 from the iconic network of Fig. 4.

If a user of the system uses a graphics input device to click on the show values icon 412, the show values icon will output a signal over the connecting line 424 (Fig. 4) which will trigger the actual values icon 416. The actual values icon 416 has been configured as a show-on-execute user panel. Since the actual values icon 416 receives an array containing the random numbers output by the random number icon 408 and collected by the sliding collector 414, when it receives an input on its sequence input from the connecting line 424, a user panel pops up as illustrated in Fig. 6.

Fig. 6 shows an example of a user panel containing a show-on-execute user panel created from the iconic network of Fig. 4. Referring now to Fig. 6, a display screen 602 contains the numeric input icon 404, the strip chart icon 410, and the show values icon 412. In the example of Fig. 6, the iconic network of Fig. 4 has processed, and the user has clicked on the show values icon 412. This causes the actual values user panel 604 to pop up on the display and show the actual random numbers which have been displayed on the strip chart 410. If the show-on-execute option had not been set for the actual values user panel 604, the icon would have processed without displaying the user panel. The show-on-execute option is only available for a user panel connected to a user object, and is not available for the main user panel connected to the entire network.

The following description of the flow charts of Figs. 7-10 will illustrate the process of dynamically creating user panels and the show-on-execute function.

Fig. 7 shows a flow chart of the top level process used by the iconic programming system to interact with the user. Referring now to Fig. 7, after entry, block 702 determines whether the user has clicked the mouse graphic input device. If not, block 702 transfers to block 704 which determines whether there has been user input, and if not, block 704 returns to block 702. This loop continues until the user either clicks the mouse device on an icon, or types input at the keyboard. When either of these occurs, control transfers to block 706 which determines whether the user is requesting that the system create a user panel. A user panel may be created for the entire iconic network or for any user objects, also called subprograms, within the network. If the user is requesting a create user panel, block 706 transfers to block 708 which calls Fig. 8 to create, or add to, the panel. Block 708 then returns to block 702 to monitor input. If the user did not request the creation of a user panel, block 706 transfers to block 710 which determines whether the user requested that a user panel be secured. If the user requested the secure user panel function, block 710 transfers to block 712 which calls Fig. 9 to secure the panel. After return from Fig. 9, block 712 transfers back to block 702 to wait for user input. If the user did not request the secure user panel option, block 710 transfers to block 714 which determines whether the user requested setting the show-on-exec option for a user panel connected to a user object. If the user requested setting the show-on-exec option, block 714 transfers to block 716 which sets the show-on-exec flag for the user panel connected to a user object and transfers to block 702 to wait for user input. If the user did not request the show-on-exec function, block 714 transfers to block 718 which processes the user request before returning to block 702 to await additional user input. The many other func-

BNSDOCID: <EP 0508633A2>

tions that can be performed with the iconic programming system by block 718 are not described in this application, however, they are described in many of the applications referred to in the background of the invention.

Fig. 8 shows a flow chart of the create, or add to, user panel process. Referring now to Fig. 8, after entry, block 802 determines whether a user panel has been previously created. If a user panel has not been previously created, block 802 transfers to block 804 which calls the windowing system to establish a new user panel. The system of the present invention is implemented on a computer system that uses a windows programming environment. An example of such a windows programming environment is the X Window System of the Unix operating system, or the MicroSoft Windows System used on the MS DOS operating system. Unix is a trademark of A T & T, and MicroSoft Windows is a trademark of MicroSoft Corporation.

After establishing a new panel, block 804 transfers to block 806 which adds the panel and detail buttons to the title bar at the top of the panel. After adding the title bar details, or if the panel had been previously created, control transfers to block 804 which makes copies of all icons selected by the user. To select these icons, the user simply uses the graphical input device to highlight an icon. In this manner, the user, or the iconic programmer, can place any desired icons from the iconic network onto the user panel. After making copies of the user selected icons, block 808 transfers to block 810 which adds these icon copies to the panel created in block 804. Block 812 then determines whether the icons fit into the user panel, and if not, transfers control to block 814 which moves icons to locations within the panel. Control then transfers to block 820 which calls the windowing system to display the entire user panel and then block 822 allows the user to distribute the icons within the panel as they desire, using the facilities available through the windowing system. After the user has performed any moving of the icons within the panel, Fig. 8 returns to Fig. 7.

Fig. 9 shows a flow chart of the secure user panel process called from Fig. 7. This process is used to set up the panel so that it cannot be altered. Referring now to Fig. 9, after entry, block 902 removes all menu access icons from all elements of the panel. Block 904 then disables interactive move and size operations of elements within the panel. In combination, this prevents further alteration of the user panel. Block 906 then determines whether the secure is permanent, and if it is, block 906 transfers to block 908 which frees all the elements of the detailed view, and block 910 removes panel and detail buttons. These two steps prevents their being reincorporated into the panel at a later date. Control then returns to Fig. 7.

Fig. 10 shows a flow chart of the execution process and illustrates the show-on-execute process for user panels for user objects. Referring now to Fig. 10, after entry, block 1002 starts the processing of the iconic network. Block 1004 then gets the next object to get processed. Block 1006 determines whether the show-on-execute flag has been set for the object, and if it has, block 1006 transfers control to block 1008 which determines whether the object has been previously processed. If the object has not been previously processed, block 1008 transfers to block 1010 which creates a carrier for the panel. A carrier is the frame around the user panel which provides a title. Block 1012 then connects the panel to the carrier and transfers to block 1014. Block 1014 calls Fig. 9 to temporarily secure the panel so that it cannot be altered while it is being displayed. After securing the panel block 1016 calls the windowing system to display the carrier and the panel.

After displaying the panel, or if the show-on-execute flag was not set, control goes to block 1018 which processes the object. Block 1020 determines whether the show-on-execute flag was set. If this flag was set, block 1020 transfers to block 1022 which unsecures and erases the panel. Control then goes to block 1024 which determines whether all objects have been processed and if not, control returns to block 1004 to get the next object. After all objects have been processed Fig. 10 returns to its caller.

The system also allows a panel for a user object to be permanently connected to a higher level panel. This allows a higher level panel to be built from the lower level panels.

Having thus described a presently preferred embodiment of the present invention, it will now be appreciated that the objects of the invention have been fully achieved, and it will be understood by those skilled in the art that many changes in construction and widely differing embodiments and applications of the invention will suggest themselves without departing from the spirit and scope of the present invention. The disclosures and the description herein are intended to be illustrative and are not in any sense limiting of the invention, more preferably defined in scope by the following claims.

Claims

1. In an iconic programming system process (122) executing in an operating environment, a method for dynamically creating user panels comprising the steps of:
   (a) accepting a panel display request (706);
   (b) if a panel has been created, continuing

BNSDOCID: <EP 0508633A2>

with step (d) (802);
(c) establishing a new panel (804);
(d) accepting user input for selecting at least one icon (808);
(e) placing copies of said at least one icon on said new panel to create said user panel (810); and (f) displaying said user panel (820).

2. The process of claim 1 wherein step (e) further comprises the step of:
(e1) if said at least one icon does not fit within boundaries of said new panel, moving said at least one icon until said at least one icon fits within said new panel boundaries (814).

3. The process of claim 1 wherein step (e) further comprises the step of:
(e1) allowing a user of said iconic programming system to move said at least one icon within boundaries of said new panel (822).

4. The process of claim 1 wherein step (e) further comprises the step of:
(e1) allowing a user of said iconic programming system to adjust the size of said at least one icon within boundaries of said new panel (822).

5. The process of claim 1 further comprising the steps of:
(g) if said user panel is connected to a user object, accepting user input to set a show-on-execute flag for said user panel (714); and
(h) when processing said iconic network, if said show-on-exec flag is set (1006), displaying said user panel when an icon within said user panel is processed (1016).

6. The process of claim 1 further comprising the steps of:
(g) if a user selects a secure panel option (710), removing all means from said user panel that would allow said user panel or the iconic programming system process to be modified (712).

7. The process of claim 1 further comprising the step of:
(g) if said user panel is connected to a user object, accepting user input to permanently connect said user panel to a higher level panel.

8. In an iconic programming system process executing in an operating environment, a method for dynamically creating user panels comprising the steps of:
(a) accepting a panel display request (706);
(b) if a panel has been created, continuing with step (d) (802);
(c) establishing a new panel (804);
(d) accepting user input for selecting at least one icon (808);
(e) placing copies of said at least one icon on said new panel to create said user panel (810);
(f) if said user panel is connected to a user object, accepting user input to set a show-on-execute flag for said user panel (714);
(g) displaying said user panel (820); and
(h) when processing said iconic network, if said show-on-execute flag is set (1006), automatically displaying said user panel when an icon within said panel is processed (1016).

9. The process of claim 8 wherein step (e) further comprises the step of:
(e1) if said at least one icon does not fit within boundaries of said new panel, moving said at least one icon until said at least one icon fits within said new panel boundaries (814).

10. The process of claim 8 wherein step (e) further comprises the step of:
(e1) allowing a user of said iconic programming system to move said at least one icon within boundaries of said new panel (822).

11. The process of claim 8 wherein step (e) further comprises the step of:
(e1) allowing a user of said iconic programming system to resize said at least one icon within boundaries of said new panel (822).

12. The process of claim 8 further comprising the steps of:
(g) if a user selects a secure panel option (710), removing all means from said user panel that would allow said user panel or said iconic programming system process to be modified (712).

13. The process of claim 8 wherein step (f) further comprises the step of:
(f1) if said user panel is connected to a user object, accepting user input to permanently connect said user panel to a higher level panel.

BNSDOCID: <EP 0508633A2>

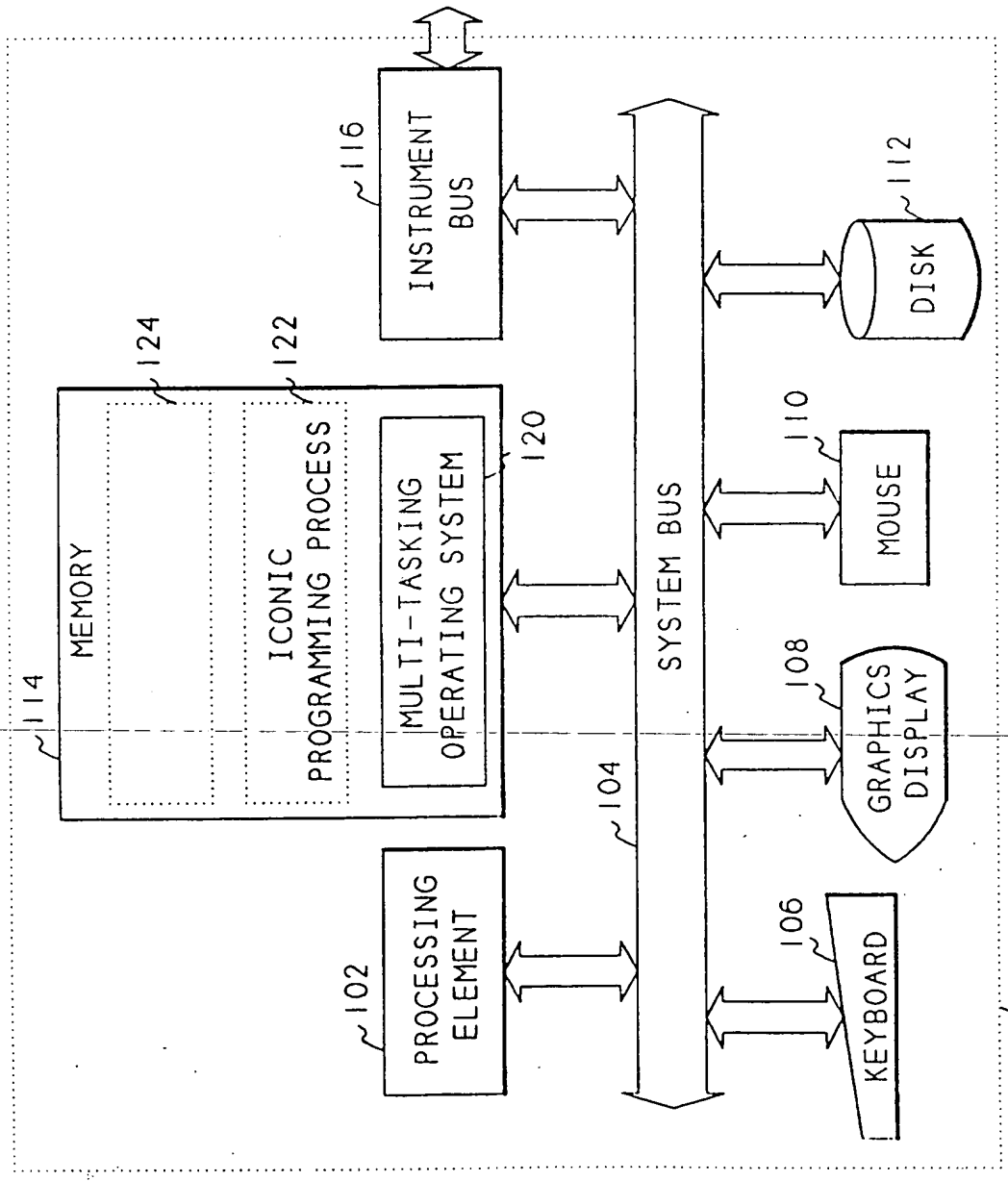

FIG. 1

BNSDOCID: <EP 0508633A2>

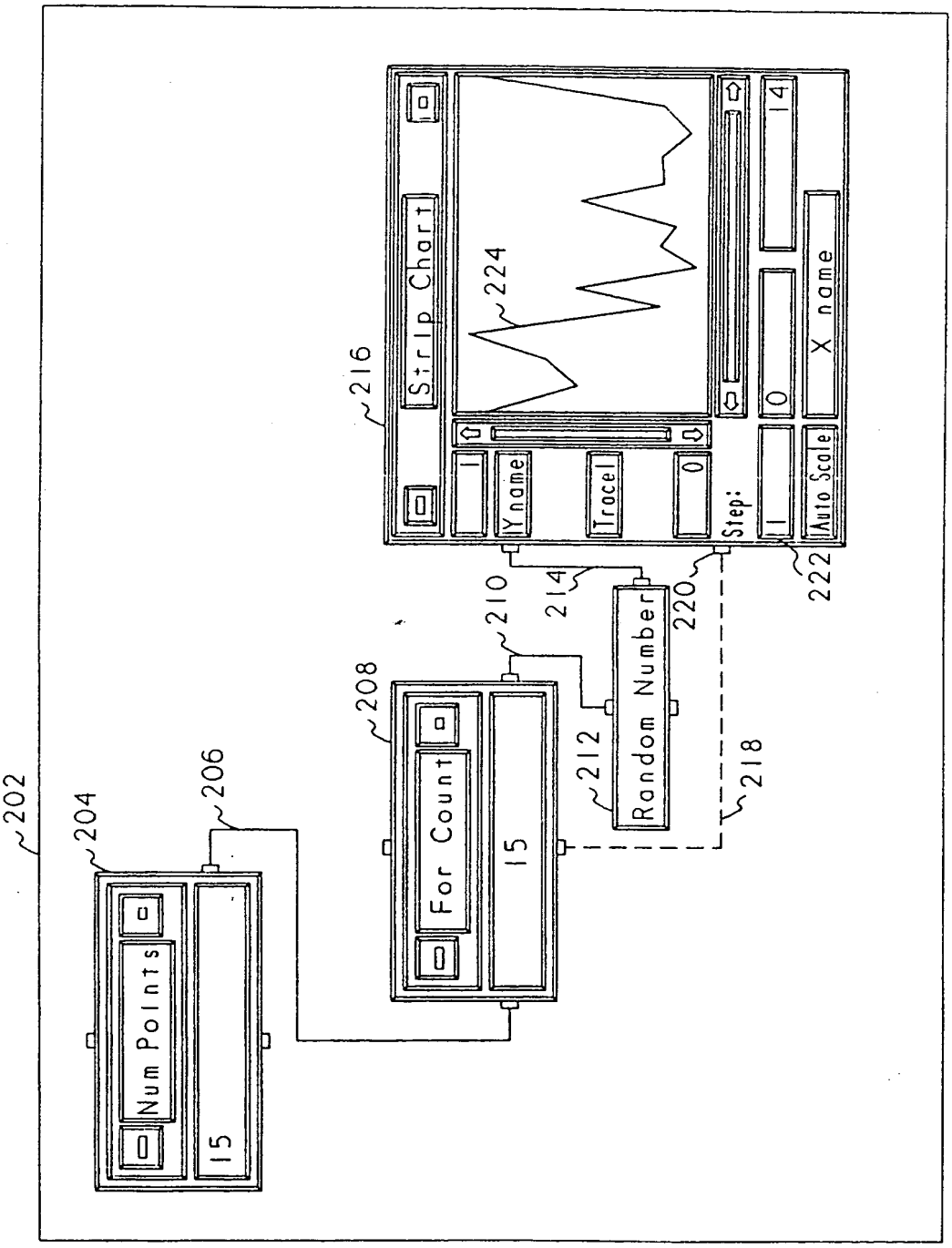
202
204
Num Points
15
206
208
For Count
15
210
212
Random Number
214
216
Strip Chart
224
Yname
Trace1
0
Step:
1
Auto Scale
10
14
X name
220
218
222

FIG. 2

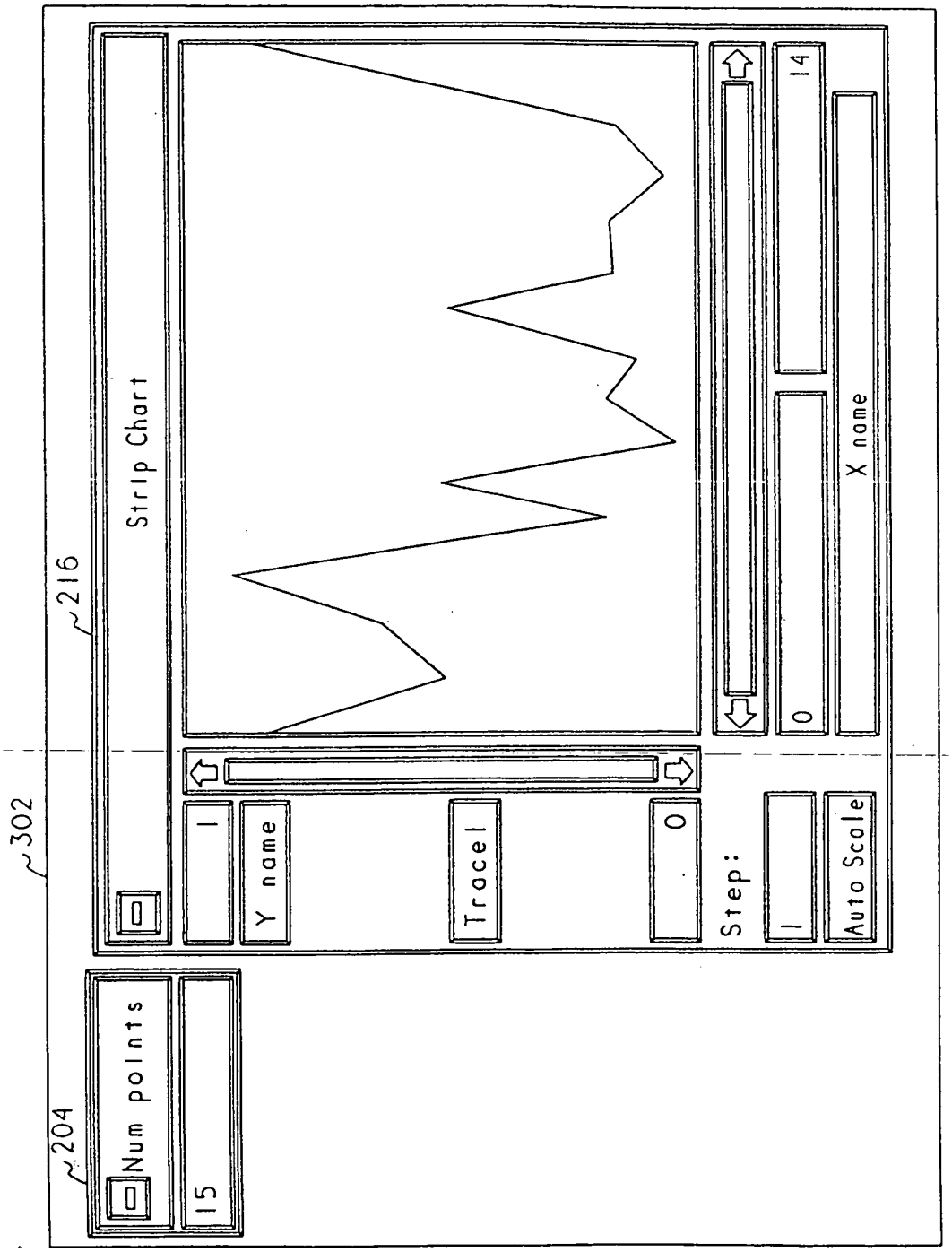

FIG. 3

BNSDOCID: <EP 0508633A2>

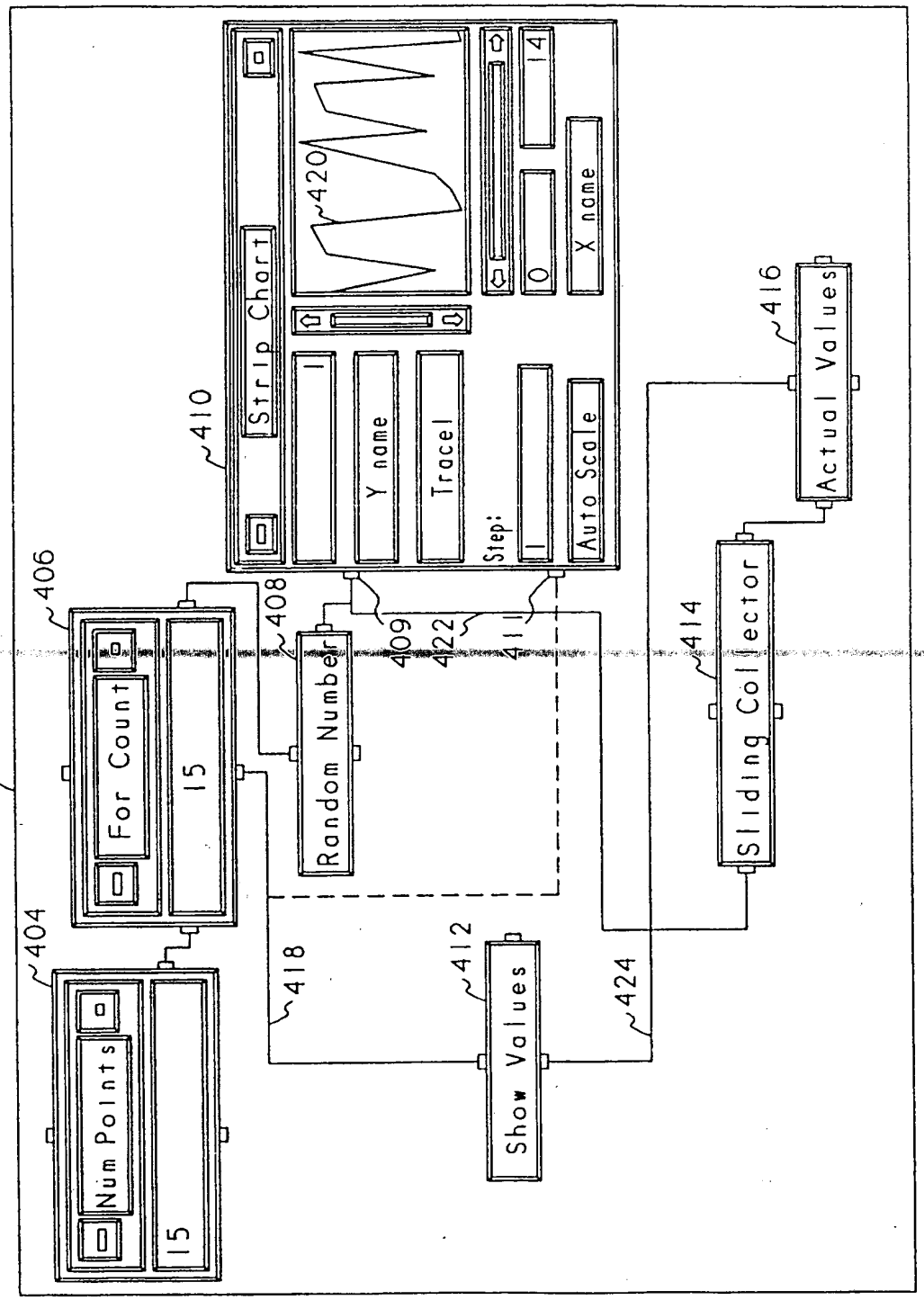
402
404
406
408
409
410
411
412
414
416
418
420
422
424
Strip Chart
Y name
Trace1
Step:
Auto Scale
X name
Actual Values
Sliding Collector
Random Number
For Count
15
Num Points
15
Show Values

FIG. 4

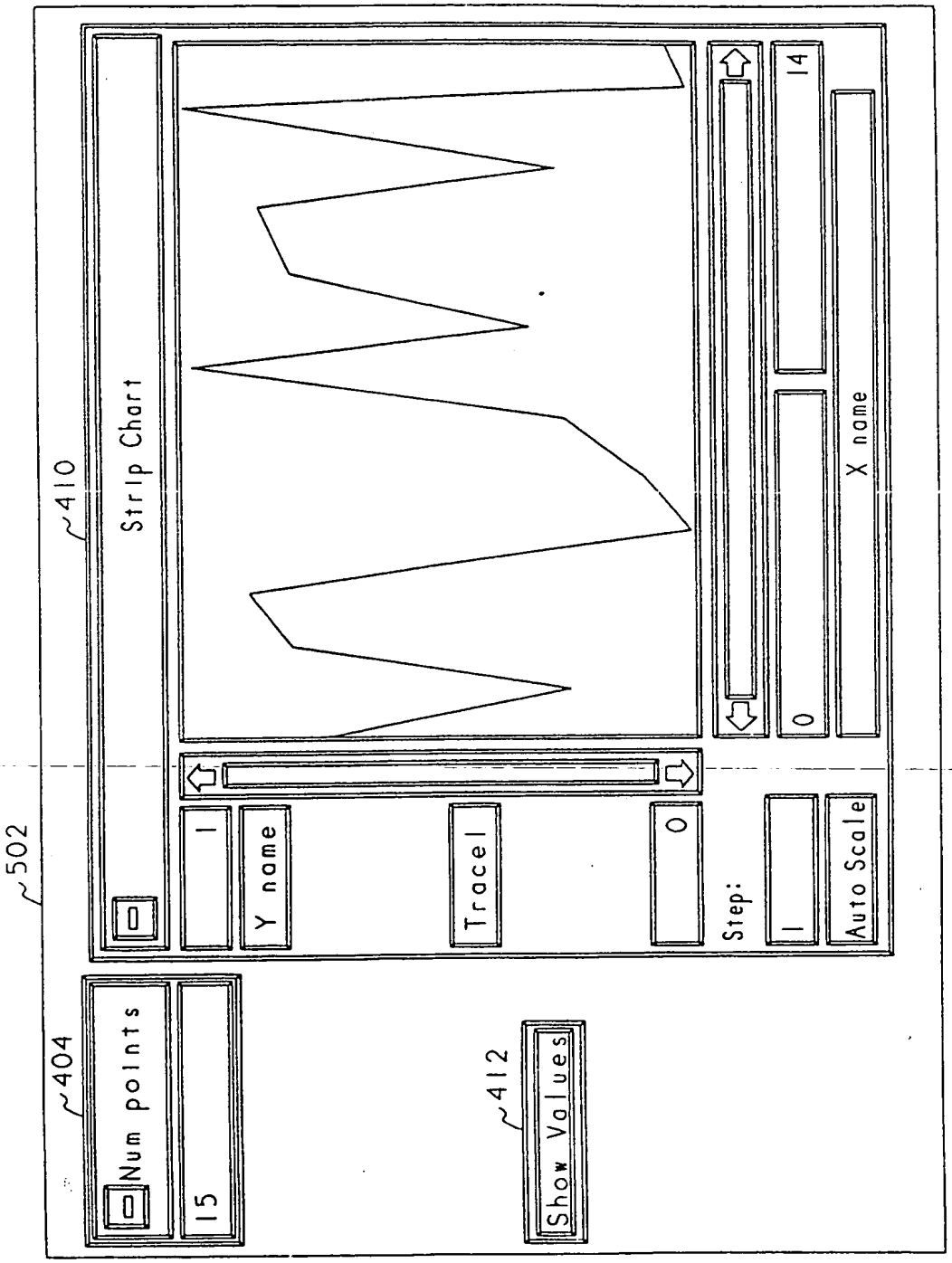

FIG. 5

BNSDOCID: <EP 0508633A2>

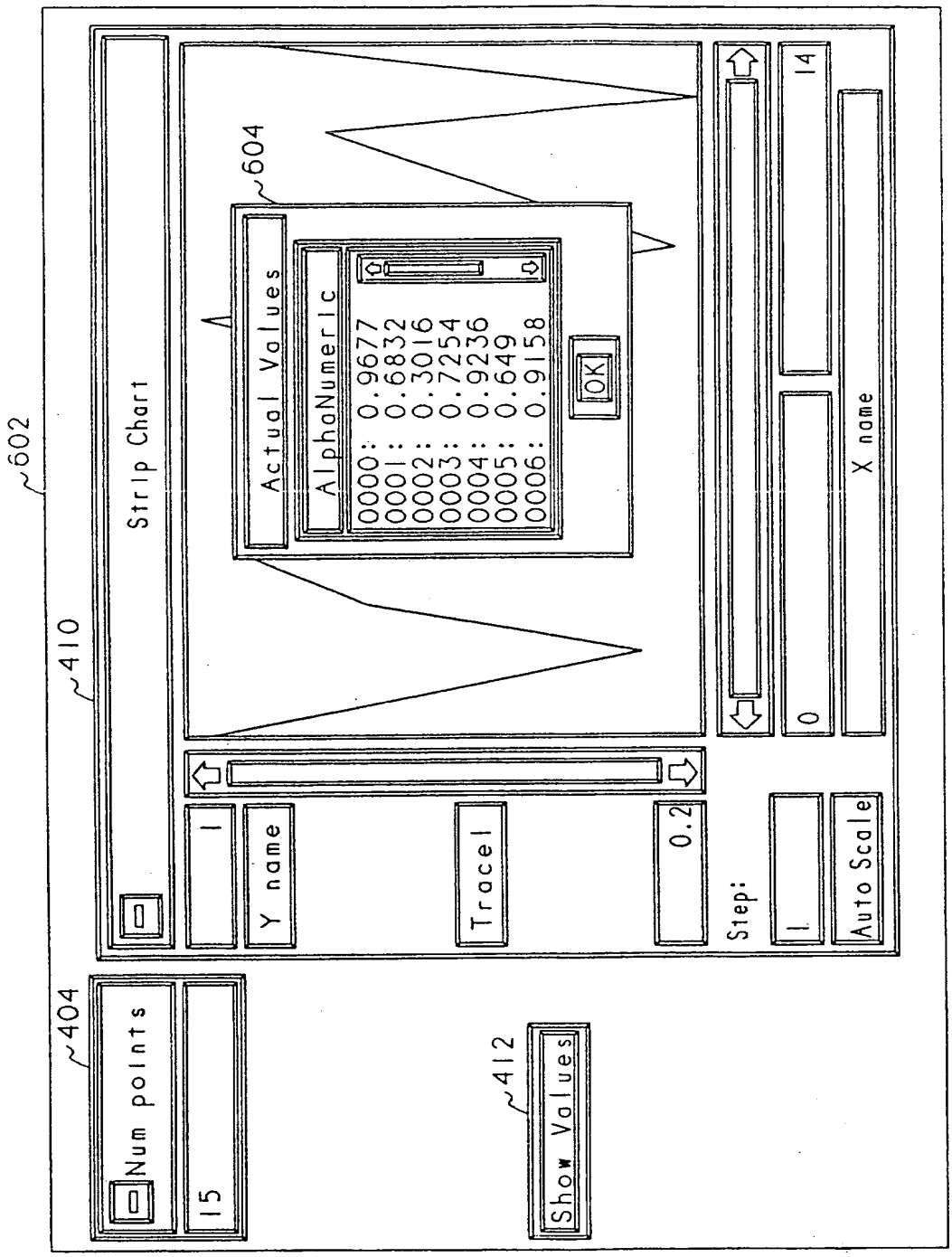
602
410
Strip Chart
604
Actual Values
AlphaNumeric
0000: 0.9677
0001: 0.6832
0002: 0.3016
0003: 0.7254
0004: 0.9236
0005: 0.649
0006: 0.9158
OK
14
0
X name
1
Y name
Trace1
0.2
Step:
1
Auto Scale
404
Num points
15
412
Show Values

FIG. 6

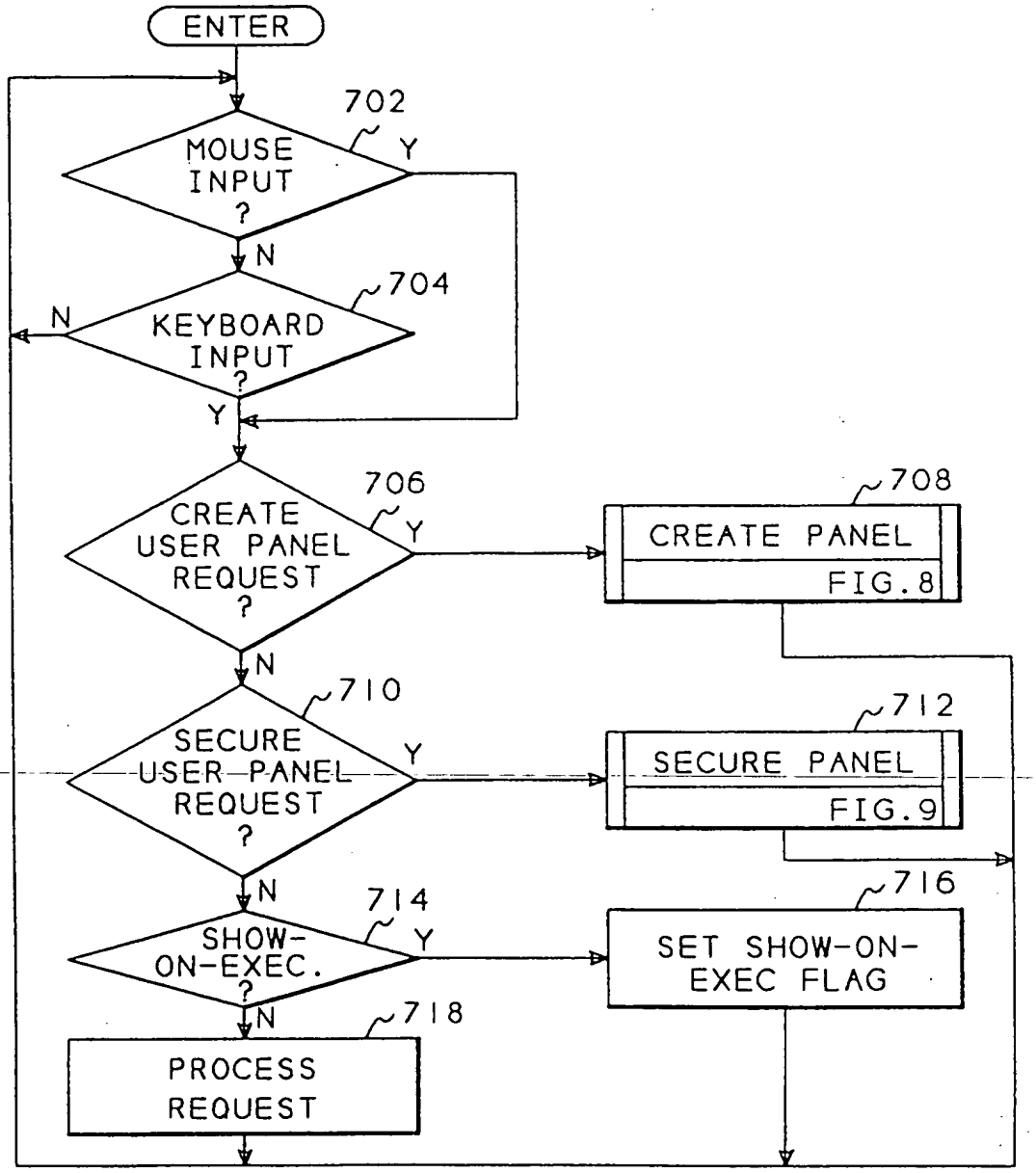

FIG. 7

BNSDOCID: <EP 0508633A2>

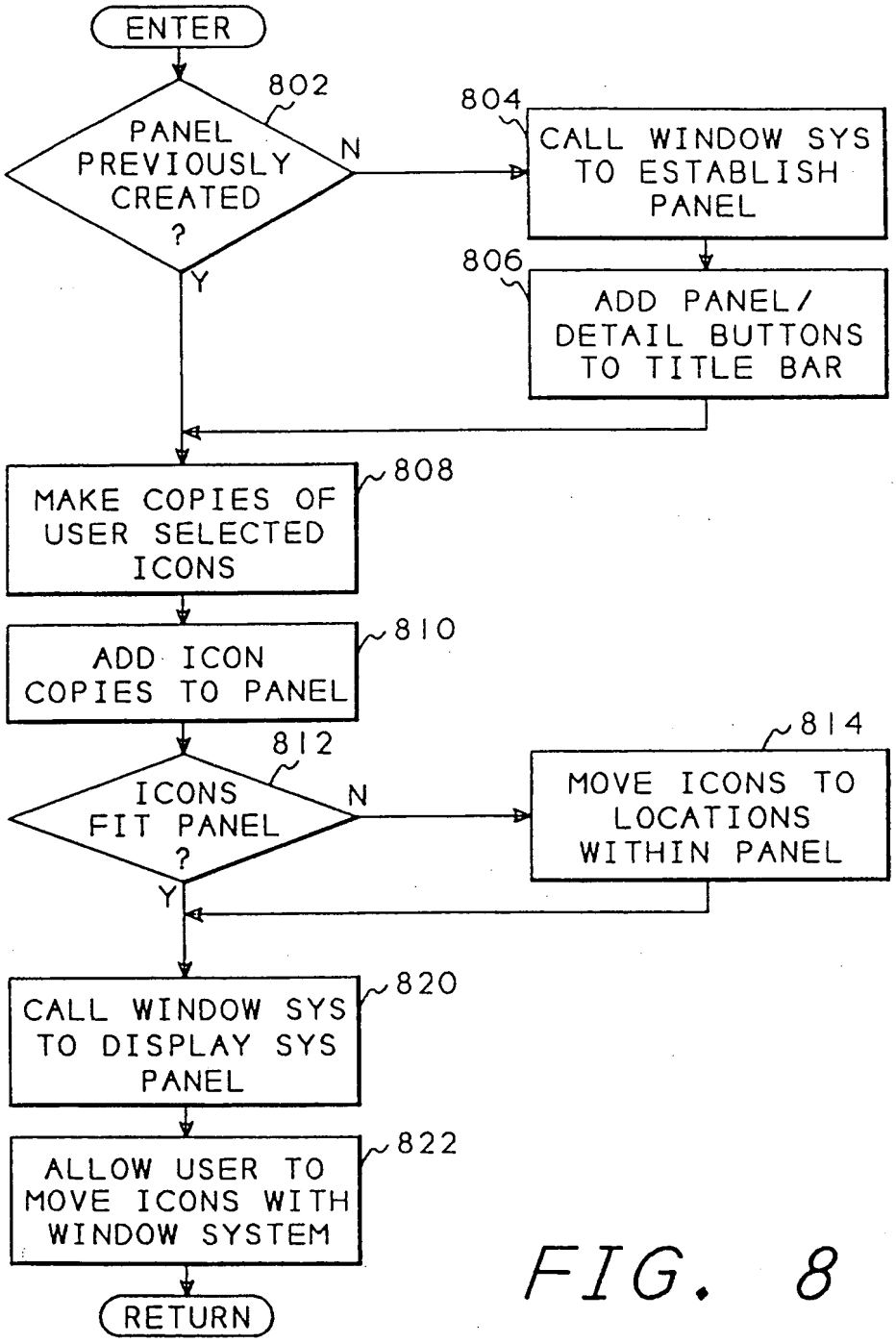

FIG. 8

BNSDOCID: <EP 0508633A2>

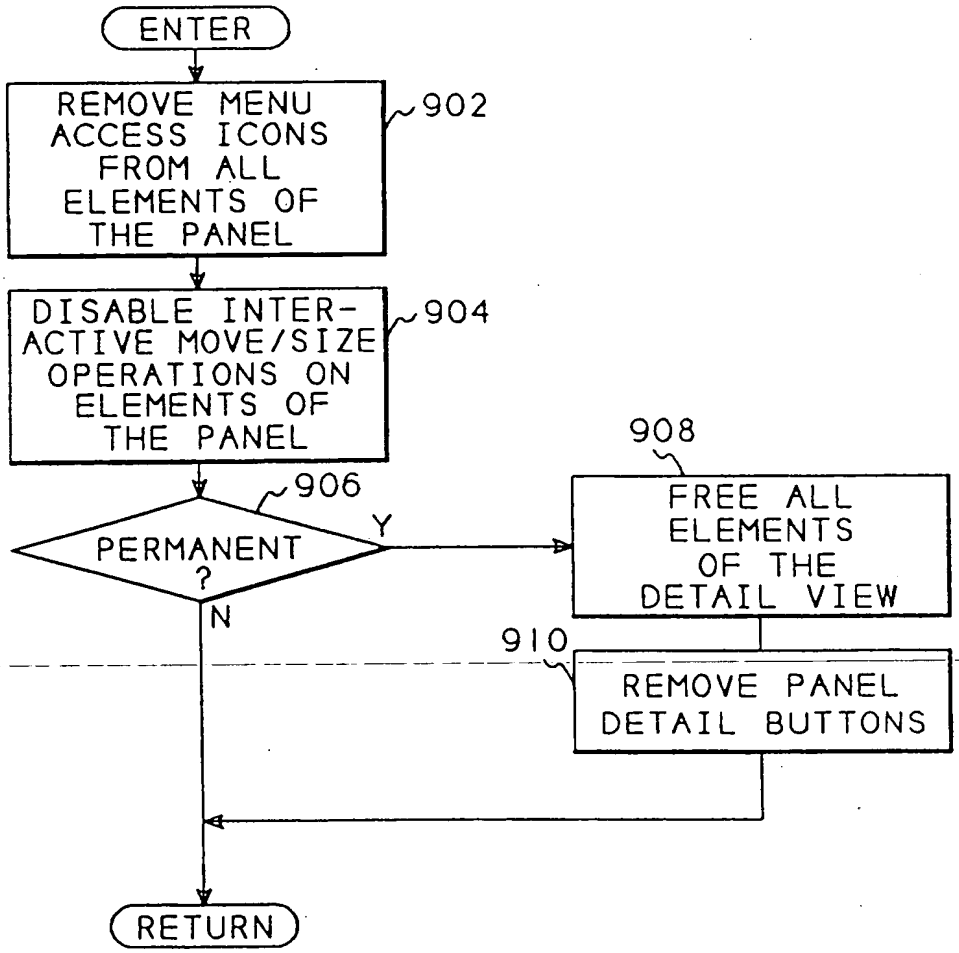

FIG. 9

BNSDOCID: <EP 0508633A2>

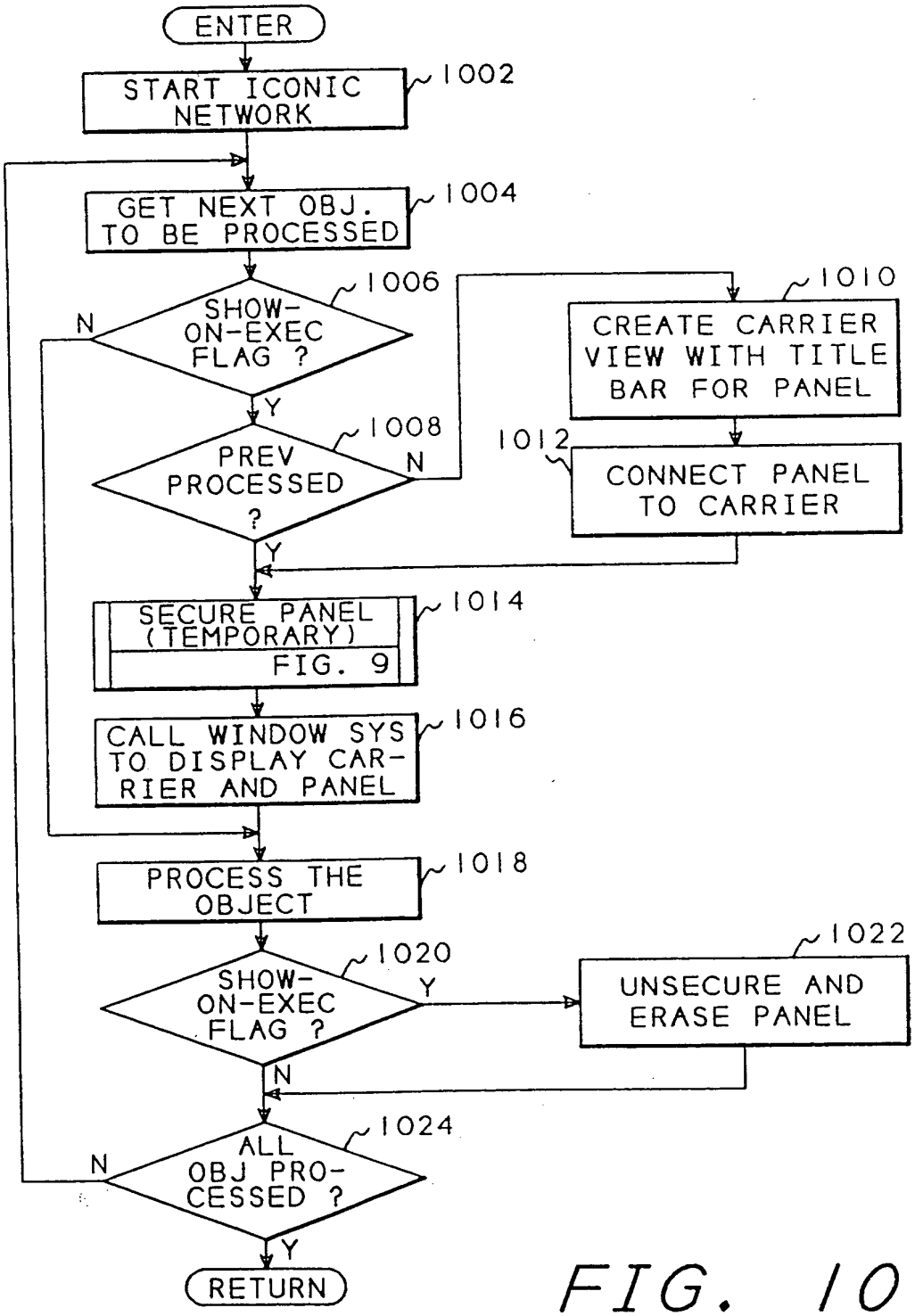

FIG. 10

BNSDOCID: <EP 0508633A2>

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 508 633 A3**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: **92302428.5**

(51) Int. Cl.5: **G06F 9/44**, G06F 3/033

(22) Date of filing: **20.03.92**

(30) Priority: **12.04.91 US 685236**

(43) Date of publication of application:
14.10.92 Bulletin 92/42

(84) Designated Contracting States:
DE FR GB

(88) Date of deferred publication of the search report:
05.10.94 Bulletin 94/40

(71) Applicant: **Hewlett-Packard Company**
3000 Hanover Street
Palo Alto, California 94304 (US)

(72) Inventor: **Hunt, William L.**
8307 Bruns Drive
Fort Collins, Colorado 80525 (US)

(74) Representative: **Colgan, Stephen James et al**
CARPMAELS & RANSFORD
43 Bloomsbury Square
London WC1A 2RA (GB)

(54) Dynamic user panels.

(57) Disclosed is a system that allows a programmer or user to dynamically create a user panel (706, 708), including selecting the icons (808), for an iconic network program, or part of an iconic network program, within an iconic programming environment. The programmer or user may select a show-on-execute option (714) for the panel which causes the panel to be displayed when any one of the icons within the panel is processed by the iconic programming system (1006, 1016). The panel may also be secured by removing all panel options that would allow any change to the panel or the iconic network program (712).

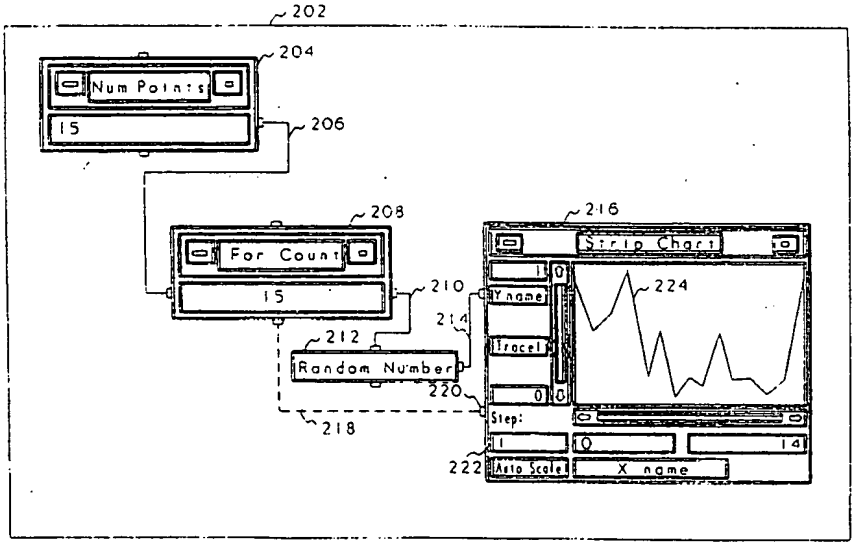

FIG. 2

EP 0 508 633 A3

BNSDOCID: <EP 0508633A3>

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 92 30 2428

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.5)
X	IEEE SOFTWARE vol. 7, no. 6 , November 1990 , LOS ALAMITOS, CA, US pages 63 - 68 T. ICHIKAWA ET M. HIRAKAWA 'Iconic Programming: Where to Go?' * page 66, column 1, line 1 - column 2, line 9 *	1-4,8-11	G06F9/44 G06F3/033
A	PROCEEDINGS OF THE ACM SIGGRAPH SYMPOSIUM ON USER INTERFACE SOFTWARE, BANFF, ALTA., CANADA, 17-19 OCT. 1988 pages 144 - 151 SMITH D. N. 'Building interfaces interactively' * page 148, left column, line 3 - right column, line 5 * * page 149, right column, line 9 - page 150, left column, line 9; figure 3 *	1-4,8-11	
			TECHNICAL FIELDS SEARCHED (Int.Cl.5) G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	11 August 1994	Bravo, P

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

BNSDOCID: <EP 0508633A3>